Received: December 29, 2023. Revised: January 19, 2024.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Enhancement of a Path-Finding Algorithm for the Hovercraft System Based on Intelligent Hybrid Stochastic Methods

Sura Muhi Hussein[1]* **Ahmed Sabah Al-Araji**[1]

[1]*Computer Engineering Department, University of Technology –Iraq, Baghdad, Iraq*
* Corresponding author's Email: 120126@uotechnology.edu.iq

Abstract: As a result of the widespread adoption of hovercraft systems, the path finding of such systems has become an essential rule for locating the target with the shortest distance and avoiding collision between the starting point and the target locations. The purpose of this paper is to develop a method for path finding by proposing a hybrid method of intelligent optimization techniques, including two stochastic approaches. The first one is called the Artificial Bee Colony (ABC) algorithm, and it aims to direct the hovercraft toward the target utilizing the behaviour of honey bees to reduce the path length required to reach the target. The second approach makes use of the Self Perception Particle Swarm Optimization (SP-PSO) algorithm, which is designed to improve particle swarm optimization in order to obtain a path that is more effective, smoother, and shorter for the hovercraft to travel in a complicated environment. The developed hybrid method is called the Artificial Bee Colony Self Perception Particle Swarm Optimization (ABC-SPPSO), which enhances the convergence speed and achieves a trade-off between exploration and exploitation in order to generate the shortest path while simultaneously avoiding collisions. The simulation results indicate that the performance of the proposed hybrid algorithm surpasses those of the original techniques. Moreover, the results demonstrate that the proposed hybrid algorithm outperforms the hybrid Firefly Algorithm Modified Chaotic Particle Swarm (HFAMCPSO) by (11.90%) in terms of distance and (56%) in terms of iterations. In addition, it outperforms the Quarter Orbit combined with the Particle Swarm Optimization (QOPSO) algorithm by (1.89%) in terms of distance and (16.66%) in terms of iterations. Furthermore, compared to the Rapid Random Tree Star Particle Swarm Optimization (RRT*PSO), the proposed method achieved enhancement of (0.82%) in terms of distance and (33.33%) in terms of iterations in providing the shortest path while avoiding collisions and reducing the trajectory tracking error to approximately zero. All the considered approaches were simulated in a global environment, utilizing the MATLAB 2022 package.

Keywords: Hovercraft system, Path finding, Static environment, Artificial bee colony, Self-perception particle swarm optimization, Trajectory tracking.

1. Introduction

Hovercrafts, sometimes referred to as air cushion vehicles (ACVs), are electromechanical vehicles with under-actuated capabilities, enabling them to traverse across many terrains, such as water, land, and diverse surfaces. In particular, the flexibility observed in these vehicles can be attributed to the presence of air cushion support, which effectively reduces friction with the underlying surface [1]. As a result, they are utilized for purposes including military, coastguard, rescue operations, transportation, searching, assisting flood victims, mine sweeping, and penetrating inaccessible regions. Due to their wide-ranging applications, it is imperative to address and resolve the challenges that impede the functionality of these vehicles [2]. When examining the applications of these vehicles, they can be categorized as Unmanned Ground Vehicles (UGVs). These vehicles require mapping, localization, and path finding for navigation [3]. In order to design a collision-free path for vehicles from their starting point to their final destination, path finding is utilized. In this regard, path finding could be categorized into two main types: offline path finding and online path finding. The distinction between

Received: December 29, 2023. Revised: January 19, 2024.

these two types is based on the degree of knowledge available about the environment. Specifically, when the vehicle has direct knowledge of all environmental data, including the trajectories of non-static objects and static obstacles, it falls under the first category. Conversely, if this data is unavailable or insufficient, the vehicle must assess the path throughout the navigation process, which is called online path finding [4, 5]. In this context, numerous researchers employed path finding for a variety of purposes using different methods to solve this problem. In this paper, to solve the hovercraft path finding issue, three objectives are achieved through the use of hybrid techniques in path finding. The initial objective was to navigate towards the target without encountering any obstacles. The minimal distance to the target is the second objective, and the third objective is the smoothest path. This intelligent hybrid methodology combines the artificial bee colony (ABC) method inspired by honey bees' intelligent behaviour with Self Perception Particle Swarm Optimization (SP-PSO) to develop the ABC-SPPSO method, which achieved the previously mentioned objectives. The structure of this paper is as follows: The hovercraft system is shown in Section 2. The types of path finding algorithms are covered in Section 3. Our hybrid approaches are explained in Section 4, and the simulation results and analysis are given in Section 5. Lastly, Section 6 represents this paper's conclusion.

2. Hovercraft system

Hovercraft systems, which are versatile, agile, and easily deployable, pose challenges in automatic control due to their under-actuated nature [6]. The concept of under-actuation refers to systems that possess a reduced number of control inputs in comparison to the number of independent generalized coordinates. This paper presents an under-actuated hovercraft featuring two control inputs and three degrees of freedom. Fig. 1 illustrates the hovercraft's configuration. The hovercraft is equipped with a pair of high-capacity ducted fans, which are responsible for both the propulsion and steering of the hovercraft. These fans provide two distinct input sources, *Fs* and *Fp*, as well as the hovercraft direction (*x*, *y*, *θ*) [7]. Hence, the formulation of the motion equations is a crucial initial step in the development of the mathematical model for the hovercraft. In particular, the kinematic equations that describe how the vehicle moves are as follows [7]:

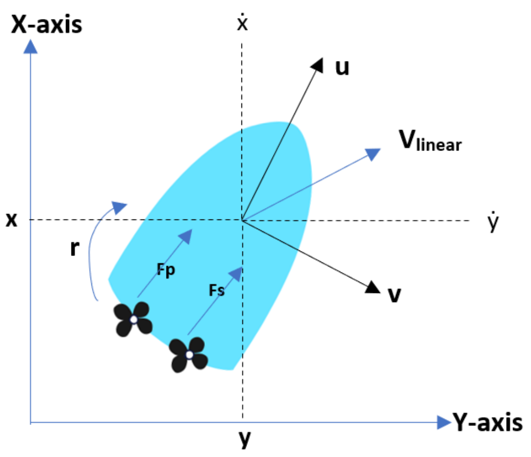


Figure. 1 Hovercraft model

$$\dot{x} = u\cos\psi - v\sin\psi \tag{1}$$

$$\dot{y} = u\sin\psi + v\cos\psi \tag{2}$$

$$\dot{\psi} = r \tag{3}$$

The dynamic equations of the hovercraft motion include Eq. (4), Eq. (5), and Eq. (6) [7].

$$m\,\dot{u} - m\,v\,r + d_v u = F_s + F_p \tag{4}$$

$$m\,\dot{v} + m\,u\,r + d_v v = 0 \tag{5}$$

$$J\,\dot{r} + d_r r = L(F_s - F_p) \tag{6}$$

The symbols representing the definitions of kinematic and dynamic equations are presented in Table 1.

Table 1. Parameters of the kinematic and dynamic equations along with their definitions

Parameter	Definition	Unit
u	Surge speed	m/sec
v	Sway speed	m/sec
x	Cartesian coordinates of its centre of mass	meter
y	Cartesian coordinates of its centre of mass	meter
ψ	Vehicle's orientation	rad
r	Angular speed	rad/sec
M	Vehicle's mass	kg
J	Rotational inertia	kg.m^2
d_v	The coefficient of viscous	kg/sec
d_r	Rotational friction	Kg.m/sec
Fs	The starboard fan force	N.
Fp	Portboard fan force	N.
L	Arm of the forces	meter

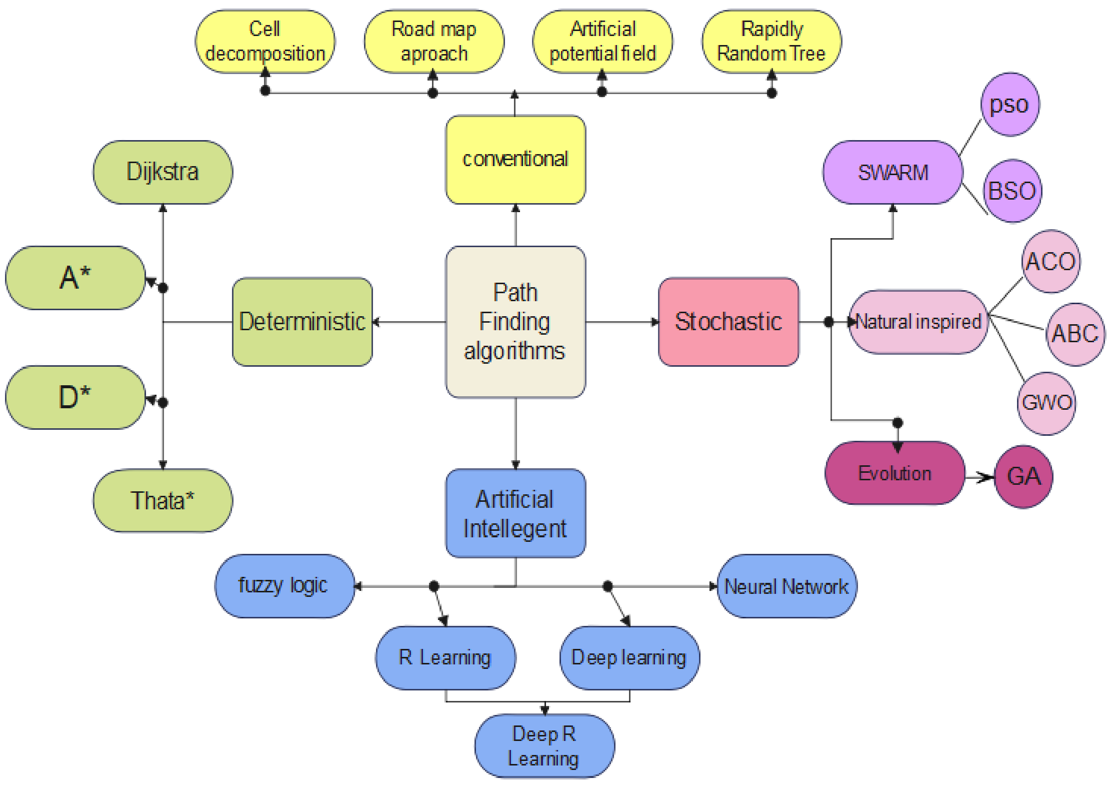


Figure. 2 Path finding algorithms

3. Path finding algorithms

A key concept in hovercraft operation is path finding, which is figuring out the best way to reach the target while taking into account things like the shortest possible path, an obstacle-free path, and the least amount of time needed to reach the target. Based on previous research works, path finding could be categorized into four main categories: Conventional, Deterministic, Stochastic, and Artificial Intelligence-based methods, as shown in Fig. 2, utilizing these techniques to locate the optimal or the near-optimal path.

3.1 Conventional methods

Conventional algorithms, developed since the 1950s, are popular in autonomous navigation due to their observable computation outputs and suitability for offline path finding in deterministic environments. In this regard, there are several conventional methods, one of which is referred to as cell decomposition (CD). The study in [8] aims to analyze the outcomes of previous studies that employed the cell decomposition technique, utilizing several criteria, such as the shortest path, minimal computing time, memory usage, safety, completeness, and optimality.

However, CD may not always be the best option for real-time applications and may struggle in more complicated contexts, such as those with dynamic impediments or ambiguous information. Another common method is known as the roadmap approach (RA) [9], which is a technique used to move between different places, and it is depicted as a sequence of one-dimensional arcs that link unoccupied areas. This strategy's performance will be significantly reduced in complex environments or higher-dimensional spaces.

In [10], Artificial Potential Field (APF) is the other most popular conventional method that generates a virtual potential field to create realistic scenes, determining vehicle trajectory based on opposition and attraction (between the obstacle and the target). However, adjusting the potential functions can be difficult, and there may be path oscillations and local minima. While in [11], the authors used a different conventional strategy by improving the RRT*(Rapid Random search Tree) algorithm to address the challenges of the winding path and the time-consuming service robot path-finding.

In particular, the primary disadvantages of these approaches are their high computational expense and the inability to execute new planning or make decisions in a dynamic environment.

Received: December 29, 2023. Revised: January 19, 2024.

3.2 Deterministic methods

Deterministic algorithms are considered problem-dependent algorithms or methods based on graphs, and their primary function is to determine the nearest neighbour [12]. One of these algorithms is the Dijkstra's algorithm used in [13] to find the shortest path in directed graphs with unbounded weights by visiting each vertex once and improving distances step-by-step, producing a shortest-path tree. Nevertheless, the computational cost becomes too high when dealing with large-scale graphs.

The second deterministic method is the A-star method proposed in 1968. In research [14], it was utilized with another method due to its efficiency and speed in discovering the shortest path, surpassing the Dijkstra's algorithm. However, the excessive number of nodes, turning points, and path smoothness made the algorithm fail to satisfy the practical motion demands. The third method [15] is the dynamic A* algorithm, also known as the D* algorithm, which is a real-time short path generation algorithm that utilizes a graph with changing or updated arc costs. Finally, the Theta* algorithm optimizes the A* path-finding approach, which is frequently used to discover the quickest route between two graph nodes. However, it causes overhead due to the additional computational work required [16].

3.3 Stochastic methods

Stochastic methods, which are problem-independent methodologies, are capable of generating solutions for an extensive array of issues. In particular, they comprise various algorithms, including particle swarm optimization (PSO) in [17], which used PSOGWO, resulting in a smoother path and overcoming the limitations of other conventional techniques. However, the PSO algorithm can also fall into local minima. On the other hand, the bat swarm optimization was utilized in [18], where Xin Yuan et al. proposed an improved bat algorithm by combining it with the dynamic window approach, producing a collision-free path, which was shorter, safer, and smoother than that produced by the standard BA. Nevertheless, the drawback was the bats' slow convergence, making it harder to acquire better optimization outcomes.

In another study [19], the ACO technique was utilized as a hybrid algorithm to solve the Travelling Salesman Problem (TSP) and obtain the shortest path. However, the convergence rate of ACO can be slow, and the memory requirements can be substantial. In [20], an enhanced GWO was proposed to alleviate the path finding difficulty that UAVs encounter in a difficult 3D environment. In [21], Karaboga and colleagues presented a comprehensive assessment of research on the Artificial Bee Colony (ABC) algorithm, covering its enhancement, hybrid approaches, and applications.

The last method in the stochastic group is the genetic algorithm (GA). In [22], the research suggested utilizing a hybrid algorithm, which was created by merging the GA and the flower pollination algorithm (FPA) to discover the most optimal route within a contemporary building's actual setting. However, this algorithm requires a lot of computing power, and the speed of completion could become slow for big problems.

3.4 Artificate intelligence methods

Recent studies showed that researchers and academics are increasingly using AI-powered methods. For example, Eyed et al. [23] presented a pathfinding method based on fuzzy logic to find the optimum path from several paths and give the best result, which is characterized by its short length and reduced processing time. However, creating suitable fuzzy rule sets can be a complex task.

Moreover, in [24], the authors explored the use of artificial neural networks (ANNs) in automatic self-navigation, discussing their structures, models, and practical applications in forecasting traffic. However, ANNs require a large amount of computer power as well as training data. As another AI-based method, deep reinforcement learning (DRL) combines deep learning and reinforcement learning. In this regard, the authors in [25] focused on deep reinforcement learning in unstructured environments. However, the limitation of this method is that it is difficult to interpret and understand the learned rules.

4. Hybrid finding algorithm

The main goal of this paper is to solve three parts of the path finding problem. The first part is to prevent the hovercraft from hitting any objects. The second part is finding the shortest hovercraft path to reach the target in a complicated environment. The third part is producing the smoothest path. To this end, a hybrid technique was presented to overcome these three challenges by combining Artificial Bee Colony (ABC) with self-perception particle swarm optimization (SPPSO) methods as a proposed hybrid stochastic method.

4.1 Artificial bee colony

The ABC algorithm, developed by Karaboga's group in 2005 [26], is a computational method that draws inspiration from the behaviour of bee colonies. This algorithm is specifically designed to handle optimization problems. Specifically, the model of honey bee colonies comprises three fundamental elements, including food sources and bees that are employed and unemployed (onlooker and scout bees), with two major forms of behaviour, namely nectar source recruitment and abandonment [27]. In this context, there are three types of bees in the ABC algorithm: recruit, onlooker, and scout bees. The recruit bees make up half of the hive, and the watching (onlooker) bees make up the other half.

The recruited bees are in charge of using the nectar sources that have already been explored and informing the other bees in the hive about the quality of the food source being used. On the other hand, onlooker bees remain within the hive. The information provided by the recruited bees is utilized to determine which food source to exploit. Meanwhile, scouts, motivated by internal motivation or external clues, randomly search the environment for new food sources to improve their ability to avoid local optimums.

According to Eq. (7) [28], a random sample of food source solutions (SN) is used to create the initial population.

$$x_{i,j} = x_{j(min)} + rand(0,1)(x_{j(max)} - x_{j(min)}) \quad (7)$$

Where x_i=1…., is the population size (SN), indicating the ith solution in the swarm, and j=1 …., D is the number of adjustable parameters.

The minimum and maximum boundaries of the solution position in dimension j are $x_{j(min)}$ and $x_{j(max)}$, respectively.

As indicated in Eq. (8), each recruit bee x_i develops a new candidate solution v_i near its initial position.

$$v_{i,j} = x_{i,j} + \varphi_{i,j}(x_{i,j} - x_{k,j}) \quad (8)$$

Where k is randomly selected from {1,…,SN}, where $k \neq i$, and φ is a random number between (0,1).

The fitness of each solution is calculated using Eq. (9):

$$fit_i = \begin{cases} \frac{1}{1+f(x_i)}, & if\ f(x_i) \geq 0 \\ 1 + |x_i|, & if\ f(x_i) < 0 \end{cases} \quad (9)$$

Where,

$$f(x_i) = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \quad (10)$$

fit_i is the solution fitness and $f(x_i)$ is the objective function (the Euclidean distance). If the new food position ($v_{i,j}$) is better than the previous one ($x_{i,j}$), the bee's memory swaps the old solution with the new one. Otherwise, the prior position is kept (a greedy selection process). After searching, the recruit bees waggle and dance to alert onlookers about their food sources. Onlooker bees choose food sources based on nectar abundance collected from recruited bees. This is done by calculating the following probability [29]:

$$Prob(i) = \frac{fit_i}{\sum_1^{SN} fit_i} \quad (11)$$

The pseudo code of the ABC algorithm is shown in Fig. 3.

```
Step1. Initialization
    Step 1.1. Randomly populate by Eq. (7)
    Step 1.2: Evaluate initial population
Repeat
Step 2. Recruited bee phase:
        Step 2.1. Create new solution Vi for the
                  recruited bee using Eq. (8)
        Step 2.2. Calculate the fitness ($fit_i$) for the
                  solution Vi using Eq. (9)
        Step 2.3. Use greedy selection between
                  old and new solutions.
Step 3. Generate the probability pi by Eq. (11).
Step 4. Onlooker bees' phase:
        Step 4.1. Select a solution to be updated
                  based on Eq. (11) then update
                  them using Eq. (8).
        Step 4.2. Calculate the fitness ($fit_i$) for the
                  updated solution.
        Step 4.3. Apply greedy selection.
Step 5. Scout bees' phase
          Step5.1. Replace abandoned scout
                   solutions with randomly
                   generated ones using Eq. (7).
Step 6. Save the best solution then set
        iteration=iteration+1, until a termination
condition is satisfied.
```

Figure. 3 Pseudocode of the ABC algorithm

Subsequently, any food source that fails to surpass a predetermined threshold of trials is discarded, and the associated recruited bee will be converted to a scout bee to generate a random location throughout the entire space using Eq. (7).

Due to its powerful global search feature, the ABC algorithm has attracted a lot of attention among experts and students.

However, there are some problems with the ABC algorithm's accuracy and slow convergence speed. Moreover, the algorithm works well for exploring but not so well for exploiting.

4.2 Self-perception particle swarm optimization

Kennedy and Eberhard created the Particle Swarm Optimization (PSO) in 1995. Particularly, the PSO is a population stochastic optimization technique inspired by fish schooling as well as birds' flocks [30], mimicking a particle swarm searching across a multidimensional search space in order to converge towards the optimal solution, with each particle representing a candidate solution.

After initializing a particular number of random particles in the population, the PSO method iteratively updates the particle swarm to find the optimal or near-optimal solution. Each particle has position and velocity that determine its state [31].

A particle's position indicates a potential solution, and its velocity regulates its progress in the search space. The particle behaviour depends on its best-known position (pbest) and the population's (gbest) [32].

However, the standard PSO suffers from the loss of population diversity and the premature convergence into a local minimum. Therefore, methods to further improve the performance of PSO have been continuously proposed.

In this regard, the suggested algorithm modifies the usual PSO algorithm by considering particle self-perception [33].

In particular, when self-perception is included, the evolution process of the whole algorithm is improved, which leads to better results in less time. The concept is derived from the capacity for self-regulated and collective learning exhibited by humans, wherein the ability to continuously assess one's performance enables individuals to develop proficiency in making more informed decisions. As the algorithm advances, particles update their positions and velocities, moving towards more favourable search regions.

The equations for updating position and velocity according to SP-PSO for the ith iteration at the nth particle in our proposed method are defined in Eq. (12) and Eq. (13), and the parameters' definitions are shown in Table 2.

Table 2. Parameters definition of SP-PSO

Symbols	Definition
$W_{i,k}$	Inertia weight of the ith particle at iteration k
$v_{i,k}$	Particle speed at iteration k
p^{SR}	The self-cognizance perception factor
p^{sp}	The social cognizance perception factor
c_1,c_2	Acceleration constants ($(c_1+c_2)<4$)
r_1,r_2	Random value between [0,1]
$L_{best,k}$	Reflects the best local position
$G_{best,k}$	Reflects the best global position
$x_{i,k}$	Current position of the ith particle at iteration k

$$v_{i,k+1} = w_{i,k}v_{i,k} + p^{SR}c_1r_1\left(L_{best,k} - x_{i,k}\right) + p^{SP}c_2r_2(G_{best,k} - x_{i,k}) \quad (12)$$

$$x_{i,k+1} = x_{i,k} + v_{i,k+1} \quad (13)$$

With these two factors, the ideas of self-regulation (SR) and self-perception (SP) are added to the normal PSO, where P^{SR}= 0 for the best particle and P^{SR}= 1 for others. P^{sp}=0 for the best particle and $P^{sp}=\Upsilon$ for others. The Υ value, ranging from 0 to 1, is determined by a threshold value derived from the confidence level of a particle. SP-PSO can reach optimal or near-optimal solutions through population interactions and information exchange.

The fundamental procedures of the SP-PSO algorithm are elucidated in Fig. 4.

4.3 Hybrid artificial bee colony self-perception particle swarm optimization (ABC-SPPSO)

Although the ABC algorithm is capable of locating a path devoid of collisions, it cannot be guaranteed to find the optimal path due to its susceptibility to improper exploitation when solving complex problems. Moreover, its sluggish convergence will result in increased power consumption. In addition, the particle swarm optimization technique can easily get stuck in local optimums, leading to poor solutions in limited paths. Therefore, in this paper, two approaches are combined to create a hybrid algorithm, ABC-SPPSO.

Received: December 29, 2023. Revised: January 19, 2024.

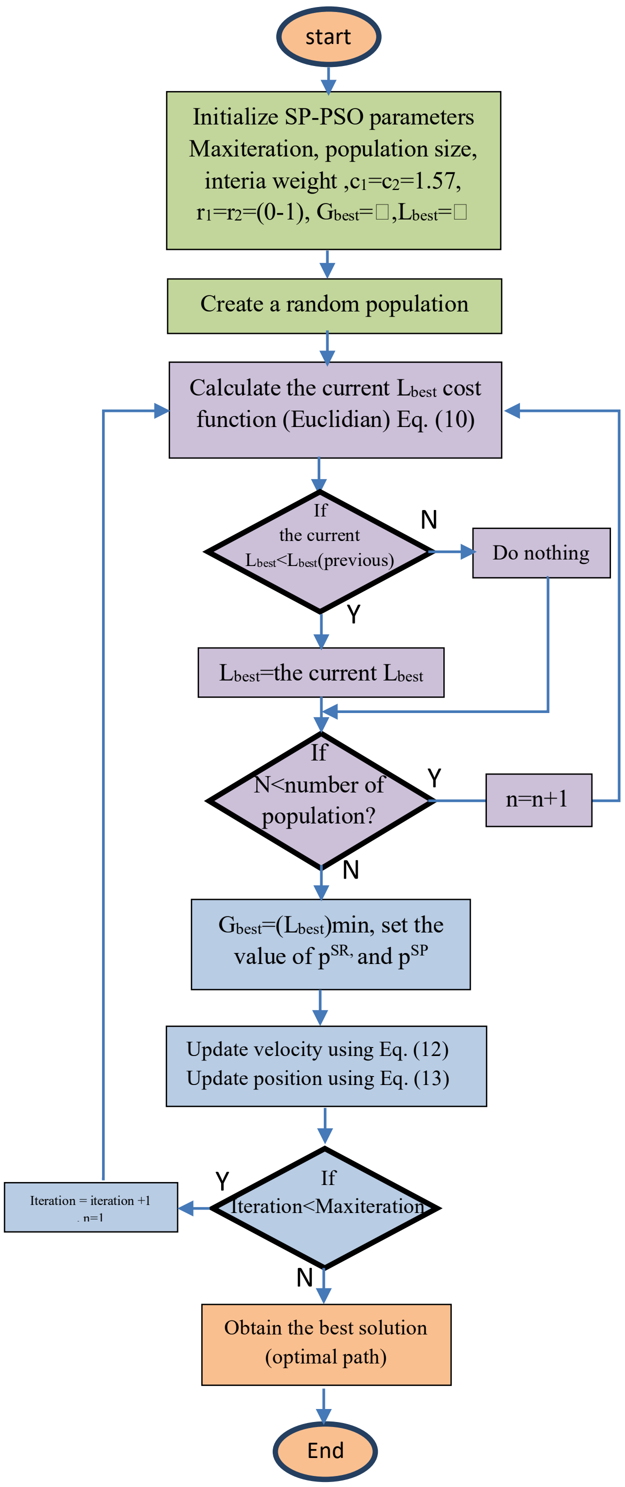

start
Initialize SP-PSO parameters Maxiteration, population size, interia weight ,c1=c2=1.57, r1=r2=(0-1), Gbest=, Lbest=
Create a random population
Calculate the current Lbest cost function (Euclidian) Eq. (10)
If the current Lbest<Lbest(previous)
N
Do nothing
Y
Lbest=the current Lbest
If N<number of population?
Y
n=n+1
N
Gbest=(Lbest)min, set the value of pSR, and pSP
Update velocity using Eq. (12)
Update position using Eq. (13)
If Iteration<Maxiteration
Y
Iteration = iteration +1 . n=1
N
Obtain the best solution (optimal path)
End

Figure. 4 The flowchart of the SP-PSO algorithm

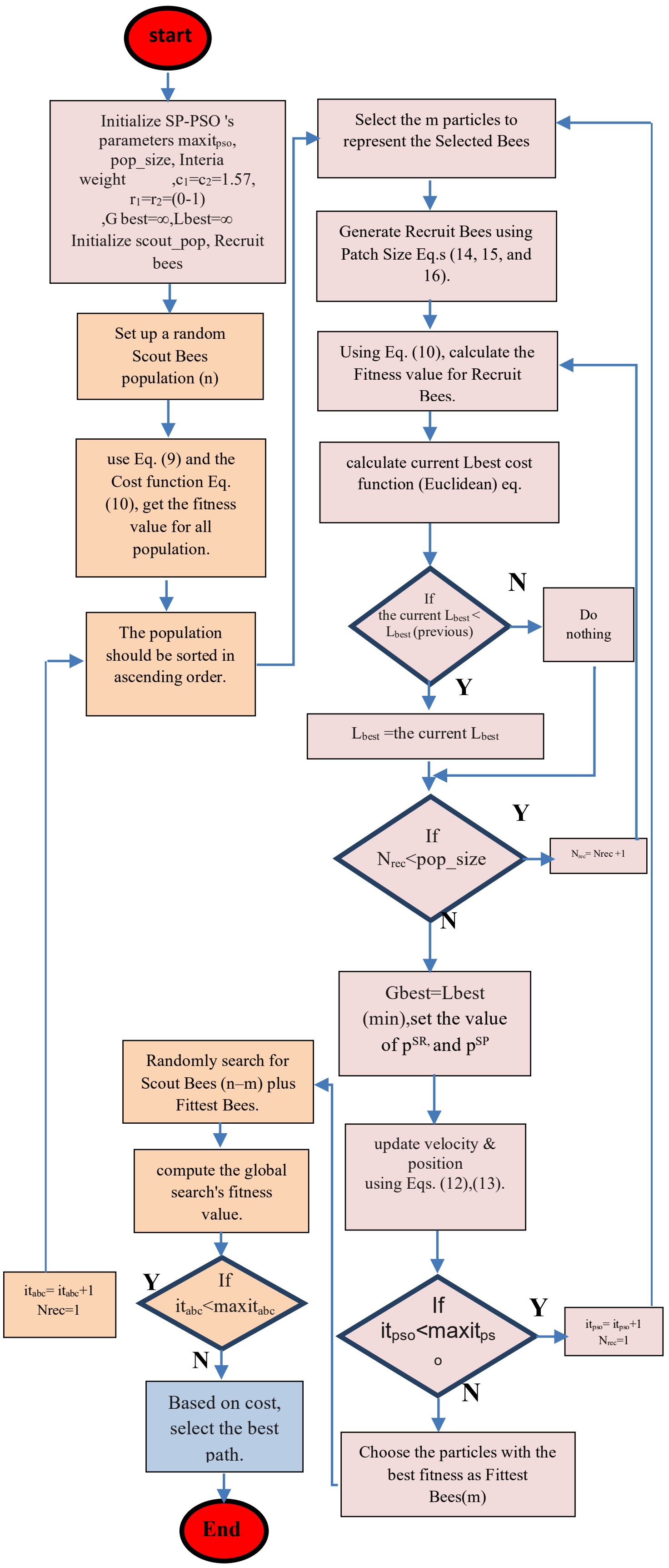

start
Initialize SP-PSO 's parameters maxitpso, pop_size, Interia weight ,c1=c2=1.57, r1=r2=(0-1) ,G best=∞,Lbest=∞ Initialize scout_pop, Recruit bees
Set up a random Scout Bees population (n)
use Eq. (9) and the Cost function Eq. (10), get the fitness value for all population.
The population should be sorted in ascending order.
Select the m particles to represent the Selected Bees
Generate Recruit Bees using Patch Size Eq.s (14, 15, and 16).
Using Eq. (10), calculate the Fitness value for Recruit Bees.
calculate current Lbest cost function (Euclidean) eq.
If the current Lbest < Lbest (previous)
N
Do nothing
Y
Lbest =the current Lbest
If Nrec<pop_size
Y
Nrec= Nrec +1
N
Gbest=Lbest (min),set the value of pSR, and pSP
update velocity & position using Eqs. (12),(13).
If itpso<maxitpso
Y
itpso= itpso+1 Nrec=1
N
Choose the particles with the best fitness as Fittest Bees(m)
Randomly search for Scout Bees (n–m) plus Fittest Bees.
compute the global search's fitness value.
If itabc<maxitabc
Y
itabc= itabc+1 Nrec=1
N
Based on cost, select the best path.
End

Figure. 5 The flowchart of the proposed hybrid (ABC-SPPSO) algorithm

Fig. 5 illustrates the concept of the hybrid methodology, which combines the ABC algorithm with the SP-PSO algorithm to accelerate convergence speed and balance between exploration and exploitation search to produce the shortest path while also preventing collisions. The ABC algorithm possesses a high level of exploration capability and requires minimal control parameters.

As a result, it ensures the discovery of a trajectory from the initial position to the target point by directing the hovercraft towards the target while avoiding obstacles until it successfully reaches the destination. Through self-perception, particles can undergo diverse modifications to facilitate quick detection and intelligent exploitation.

Consequently, employing both approaches yields an optimal or near-optimal path.

$$x_{new} = x_{old} + \alpha \times x_{old} \times random(0,1) \quad (14)$$

$$y_{new} = y_{old} + \alpha \times y_{old} \times random(0,1) \quad (15)$$

$$\psi_{new} = \psi_{old} + \alpha \times \psi_{old} \times random(0,1) \quad (16)$$

Where α is between 0 and 1.

5. Simulation results and analysis

This paper examined the effectiveness of the proposed hybrid algorithms utilizing the MATLAB program (2022) in a controlled environment. The environment consists of fixed obstacles of different shapes and a map size of [500×500] cm in both x and y axes, as depicted in Fig. 6. The computer's technical specifications include an Intel Core i5-5200U processor with 8.00 GB of RAM, and a CPU of 2.20GHz. In order to identify a path that is free from collisions, three algorithms will be employed: Artificial Bee Colony, SP-PSO, and the proposed Hybrid ABC-SPPSO algorithm. The objective is to determine the shortest-distance path by comparing the results obtained from these algorithms, taking into account that a safe distance must be maintained between the hovercraft and the obstacles.

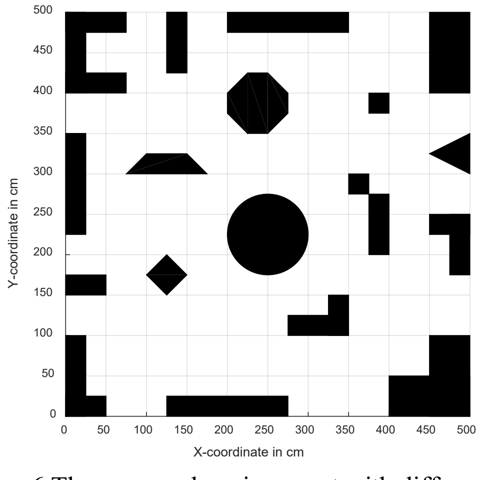


Figure. 6 The proposed environment with different fixed obstacles

Table 3. The proposed parameters of the hybrid ABC-SPPSO algorithm.

Parameters	Values
Scout Bees numbers	10
Selected Bees numbers	5
Recruit Bees numbers equal to population size of SPPSO	40
Particle numbers	5
Fittest Bees numbers	5

Table 3 shows the set of Hybrid ABC-SPPSO algorithm parameters that has been used in the simulation.

5.1 Scenario 1

The hovercraft's starting point is located at coordinates (50, 350) cm, shown by a red triangle. The target position, represented by a blue diamond, is located at coordinates (350, 450) cm. Using the artificial bee colony technique, the shortest path was 570.87cm at iteration 45. Figure 7 (a) displays the outcome of the ABC algorithm in terms of the path taken, whereas Fig. 7(b) illustrates the fitness value of the path at each iteration.

Fig. 8(a) illustrates the SP-PSO algorithm used to identify the shortest-distance path in the specified environment. As demonstrated in Fig. 8(b), the optimal cost solution was obtained at iteration 28 with a maximum of 50 iterations. The value of the SP-PSO distance function is 732.46 cm.

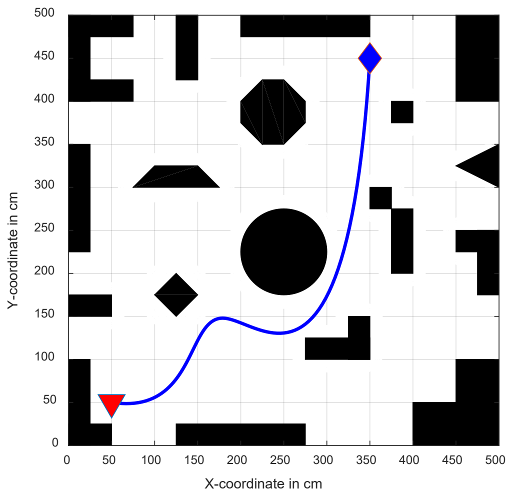
In Fig. (9) the proposed hybrid algorithm was applied to find the shortest distance path using the suggested environment, finding the best cost solution at iteration 11 with a distance function of 555.36 cm. Compared to artificial bee colony and SP-PSO algorithms, it generated a smooth path, which is the shortest. Table 4 summarizes all the outcomes of scenario 1 comparison. Table 5 summarizes the enhancement of ABC-SPPSO over the original algorithms of scenario 1 comparison.

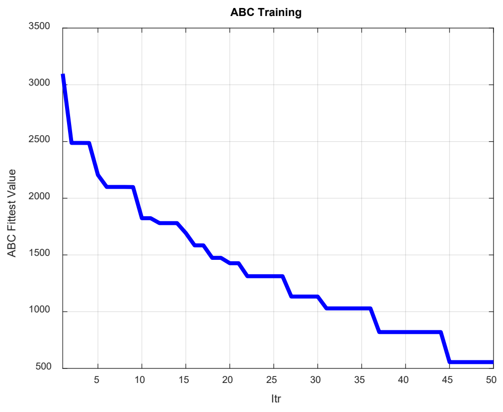
Table 4. The shortest path comparisons

Type of algorithm	Length of path	Iteration of best path
ABC	570.87cm	45
SP-PSO	732.46cm	28
Proposed hybrid ABC-SPPSO	555.36cm	11

Table 5. Enhancement comparison for scenario 1

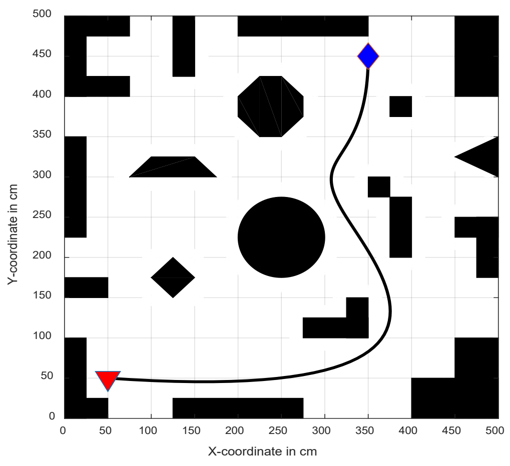
Type of algorithm	Enhancement in distance	Enhancement in iteration
ABC	2.71%	75.55%
SP-PSO	24.17%	60.71%

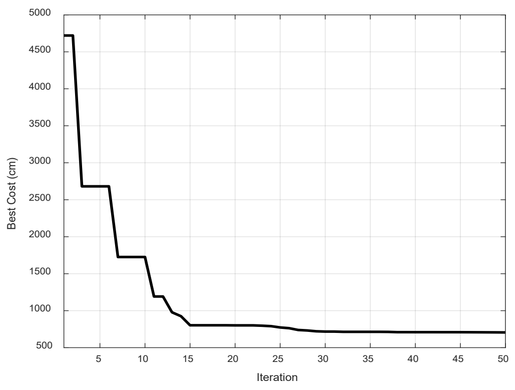




(a) (b)

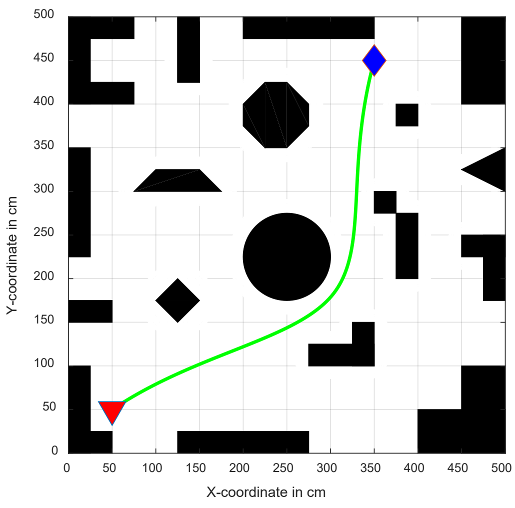
Figure. 7 The ABC algorithm: (a) path result and (b) cost function

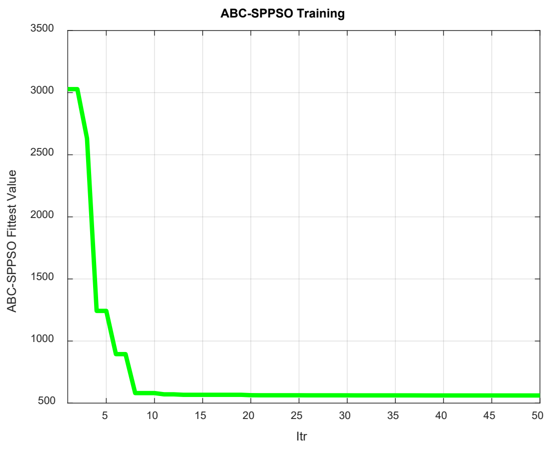




(a) (b)

Figure. 8 The SP-PSO algorithm: (a) path finding and (b) cost function





(a) (b)

Figure. 9 The ABC-SPPSO algorithm: (a) path finding and (b) cost function

Eq. (16) represents the reference path equation for the optimal path of the hybrid ABC-SPPSO.

$$Y_{ref}\left(X_{ref}\right) = 1.4305e-11 \times X_{ref}^{6} - 1.3195e-8 \times X_{ref}^{5} + 4.6824e-6 \times X_{ref}^{4} - 0.0008 \times X_{ref}^{3} + 0.067 \times X_{ref}^{2} - 1.9213X_{ref} + 40 \quad (16)$$

The inverse dynamic model of the nonlinear hovercraft is used to produce the optimal torque actions of the hovercraft to follow the reference path to determine the linear velocity V_{linear} of the hovercraft's platform, in accordance with the previous reference path, as given in Eq. (17):

$$V_{linear} = \sqrt{u^2 + v^2} \quad (17)$$

Where u and v are the surge and the sway speeds of the hovercraft, and they can be fined using Eq. (18) and Eq. (19):

$$u = \dot{Y}_{ref} \times \cos(r) - \dot{X}_{ref} \times \sin(r) \quad (18)$$

$$v = \dot{X}_{ref} \times \cos(r) + \dot{Y}_{ref} \times \sin(r) \quad (19)$$

where $\dot{x}$ represents the hovercraft's velocity along the x-axis, and $\dot{y}$ signifies its velocity along the y-axis. By utilizing Eq. (20), the angular velocity of the hovercraft (r) orientation is computed. From the dynamic equations of the hovercraft, Eq.s (21) and (22) compute the starbored (F_s) and portbored (F_p) hovercraft' fan forces.

$$r = \frac{-\dot{v}}{u} \quad (20)$$

$$F_s = 0.5 \times (M \times \dot{u} - M \times v \times \dot{r} + (J/L) \times \ddot{r}) \quad (21)$$

$$F_p = 0.5 \times (M \times \dot{u} - M \times v \times \dot{r} - (J/L) \times \ddot{r}) \quad (22)$$

Where the dimensions of the hovercraft design are 25.4 cm in depth, 35.6 cm in width, and 18.1 cm in height, and the remaining model parameters of the hovercraft are shown in Table 6 [7].

Table 6. Values of the hovercraft parameters [7]

Parameters	Values
M	5.15 kg
J	0.047 kg m^2
L	0.123 m
d_v	4.5 kg/sec
d_r	0.41 kg m/sec

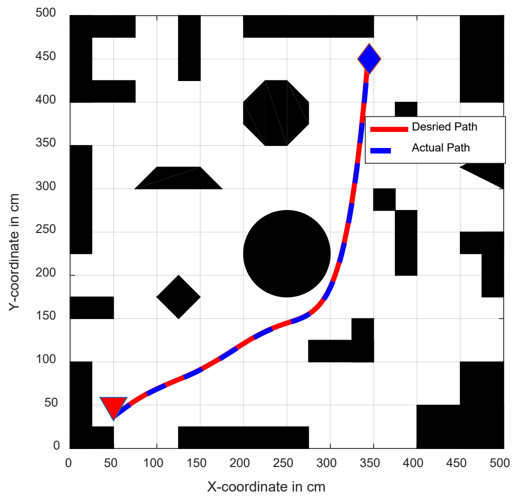
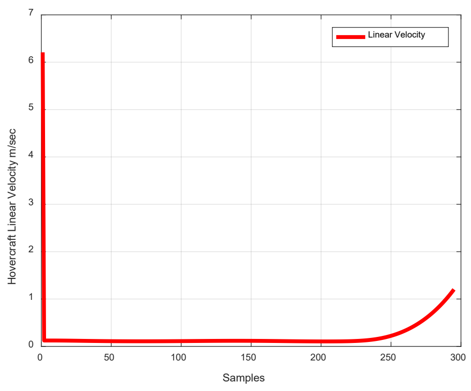


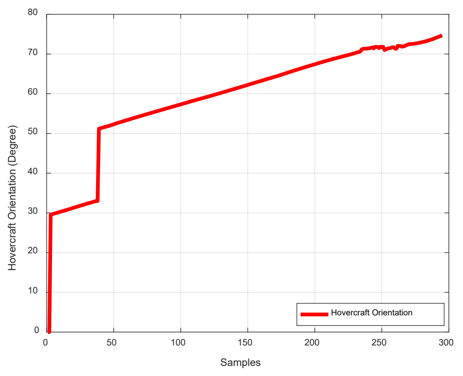
Figure. 10 The desired and the actual paths

Based on the reference path of Eq. (16), Fig. 10 illustrates how the actual path is identical to the desired path.

The smooth linear velocity of the hovercraft is depicted in Fig. 11 (a), whereas the orientation of the hovercraft is depicted in Fig. 11(b).

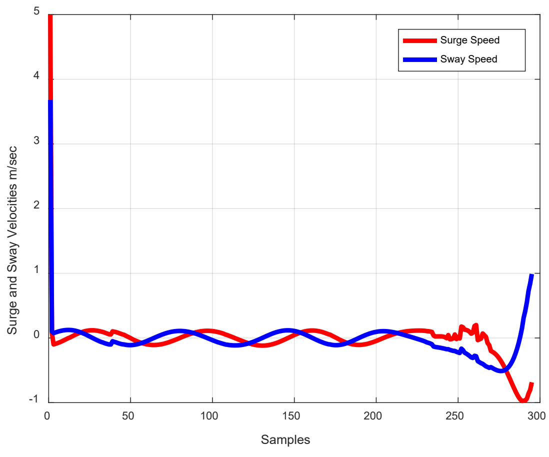


(a)

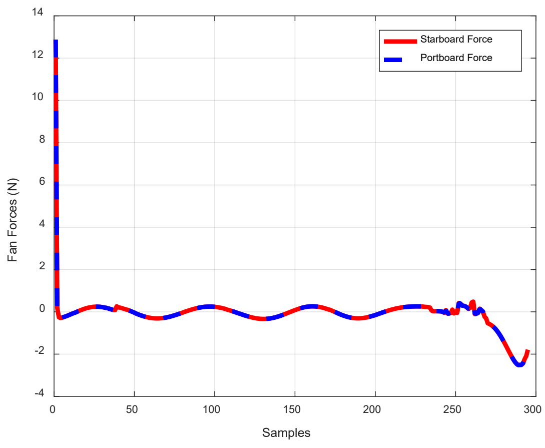


(b)

Figure. 11 Scenario 1 result: (a) linear velocity of the hovercraft and (b) the hovercraft orientation



(a)



(b)

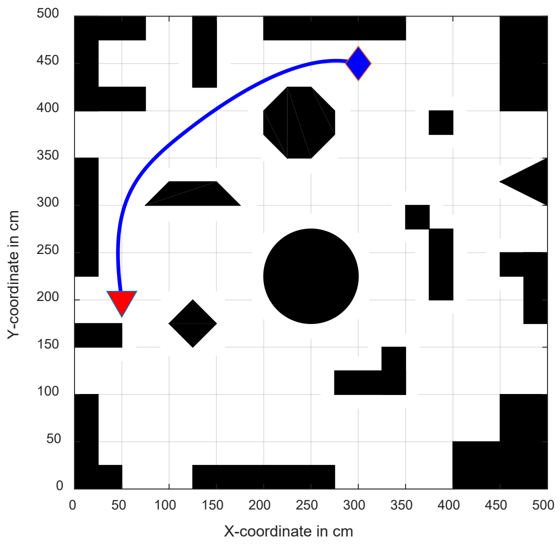
Figure. 12 Scenario 1 result: (a) surge and sway velocities of the hovercraft and (b) starboard and portboard fan forces.

Furthermore, the fast surge and the sway velocities of the hovercraft are illustrated in Fig. 12 (a), and Fig. 12 (b) illustrates the smooth response without saturation state starboard and the portboard forces of the hovercraft.

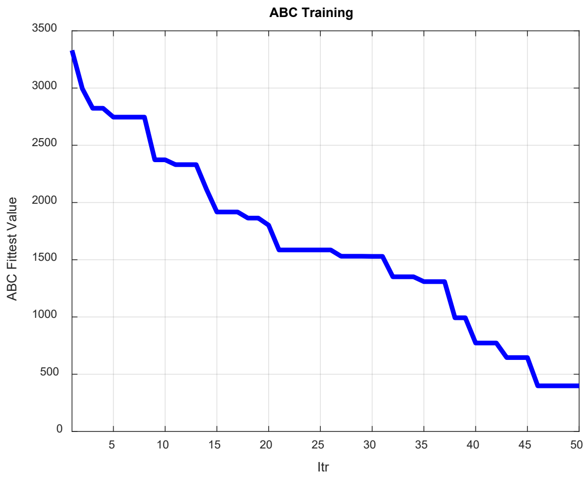
5.2 Scenario 2

In this scenario, we conducted a comparative evaluation of the proposed algorithm against the ABC and the SP-PSO algorithms. The evaluation involved using different initial points (shown by the red triangle) located at (50,200) cm, and a goal point (represented by the blue diamond) located at (300,450) cm. These evaluations were performed in the same environment in Fig. 6.

The resulting path obtained by employing the ABC method is depicted in Fig. 13 (a), while the cost function is illustrated in Fig. 13 (b). The best outcome for the cost function is achieved at iteration 46, with a value of 398.48 cm. However, when the



(a)



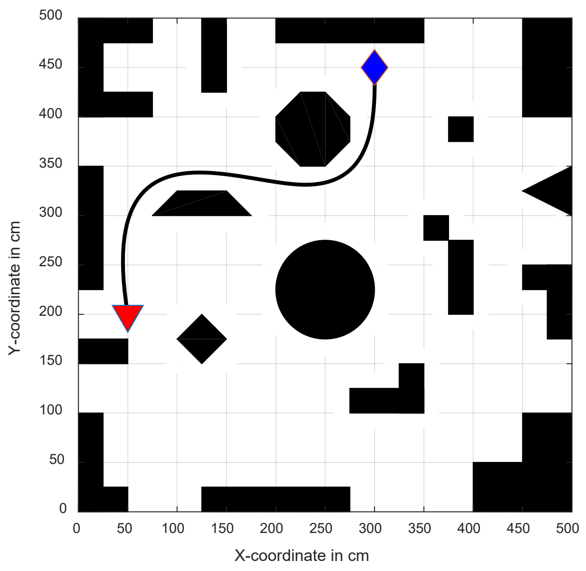
(b)

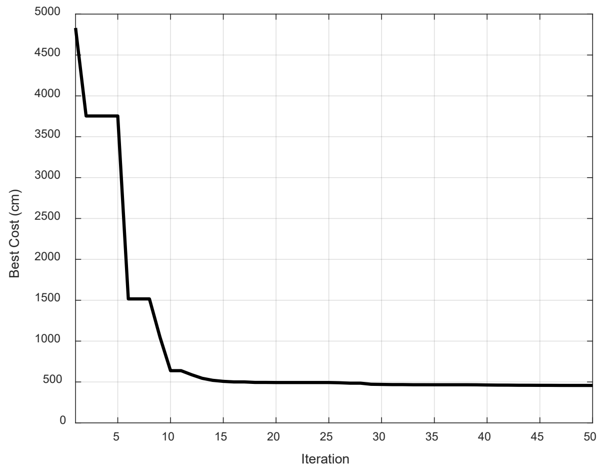
Figure. 13 Scenario 2 simulation result of the ABC algorithm: (a) the shortest path and (b) the cost function

SP-PSO is applied to the same environment, the hovercraft takes a different path, as depicted in Fig. 14 (a). The cost function of 507.47 cm at iteration 15 is shown in Fig. 14 (b).

The proposed hybrid algorithm was utilized to determine the shortest-distance path in the recommended environment, as depicted in Fig. 15 (a). The optimal cost solution was identified at the 10th iteration, as depicted in Fig. 15 (b). The maximum number of iterations was set to 50.

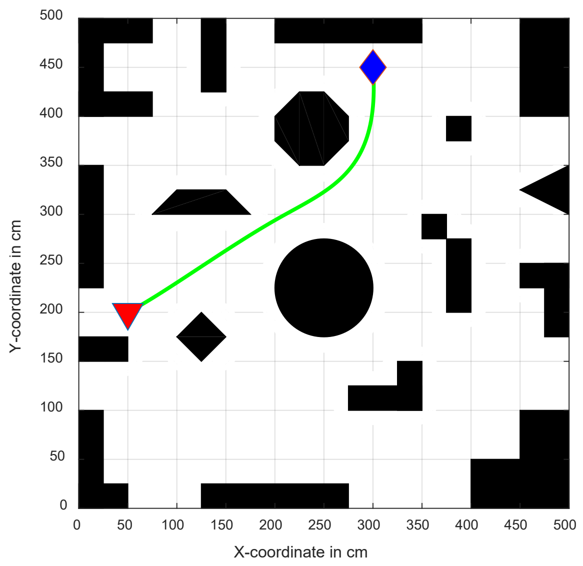
The hybrid algorithm's distance function has a value of 390.46 cm. Compared to the ABC and the SP-PSO algorithms; the path produced by the suggested hybrid algorithm was the smoothest and shortest path from the initial location to the destination point, as demonstrated in Table 7.

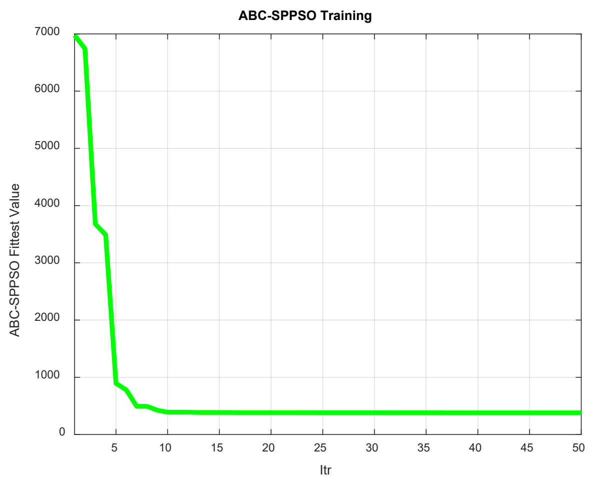




(a) (b)

Figure. 14 Scenario 2 simulation result of the SP-PSO algorithm: (a) the shortest path and (b) the cost function





(a) (b)

Figure. 15 Scenario 2 simulation result of the ABC-SPPSO algorithm: (a) the shortest path and (b) the cost function

The proposed hybrid algorithm was utilized to determine the shortest-distance path in the recommended environment, as depicted in Fig. 15 (a). The optimal cost solution was identified at the 10th iteration, as depicted in Fig. 15 (b). The maximum number of iterations was set to 50.

The hybrid algorithm's distance function has a value of 390.46 cm. Compared to the ABC and the SP-PSO algorithms; the path produced by the suggested hybrid algorithm was the smoothest and shortest path from the initial location to the destination point, as demonstrated in Table 7.

Table 8 summarizes the enhancement of the ABC-SPPSO over the original algorithms for scenario 2 comparison by using Eqs. (23 and 24).

$$\mathit{Distnace\ Enhancement}\ (\%) = \left(1 - \frac{\mathit{Proposed\ Method\ Distance}}{\mathit{Other\ Method\ Distance}}\right) \times 100\% \tag{23}$$

$$\mathit{Iteration\ Enhancement}\ (\%) = \left(1 - \frac{\mathit{Proposed\ Method\ Iteration}}{\mathit{Other\ Method\ Iteration}}\right) \times 100\% \tag{24}$$

Table 7. Comparison results for scenario 2

Type of algorithm	Best path length	Iterations
ABC	398.48cm	46
SP-PSO	507.47cm	15
Proposed hybrid ABC-SPPSO	390.46cm	10

Table 8. Enhancement comparison for scenario 2

Type of algorithm	Enhancement in distance	Enhancement in iteration
ABC	2.01%	78.26%
SP-PSO	23.05%	33.33%

Eq. (25) represents the reference path equation for the optimal path of the hybrid ABC-SPPSO for scenario 2.

$$Y_{ref}\left(X_{ref}\right) = 3.2290\mathrm{e}-11 \times X_{ref}^{6} - 3.1256\mathrm{e}-8 \times X_{ref}^{5} + 1.2014\mathrm{e}-5 \times X_{ref}^{4} - 0.002326 \times X_{ref}^{3} + 0.23663 \times X_{ref}^{2} - 11.193756X_{ref} + 394.64 \quad (25)$$

For scenario 2, the desired and the actual paths of the hybrid ABC-SPPSO algorithm are illustrated in Fig. 16, while the hovercraft's linear velocity is represented in Fig. 17 (a). On the other hand, Fig. 17 (b) displays the hovercraft orientation.
Additionally, Fig. 18 (a) displays the hovercraft's surge and sway velocities, while Fig. 18 (b) shows the starboard and the portboard forces acting on the hovercraft.

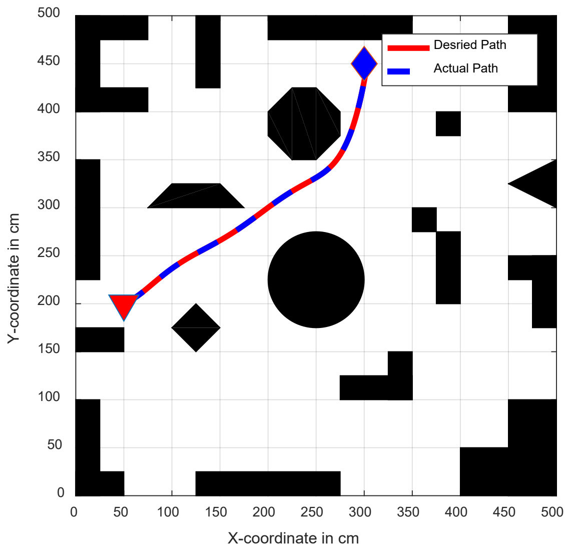
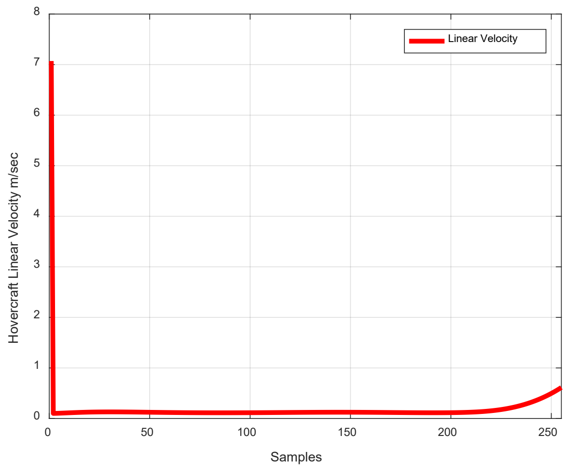
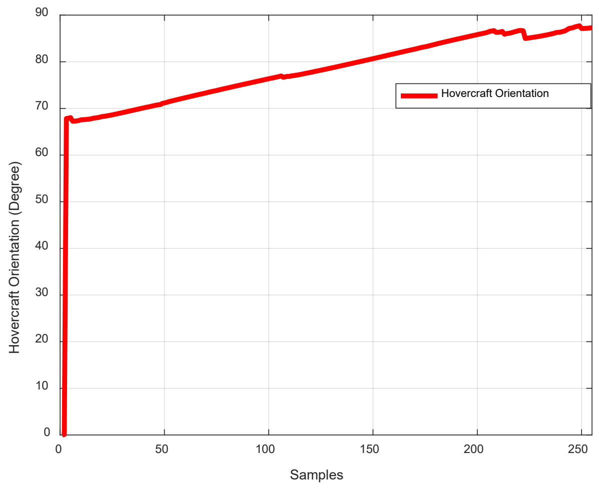


Figure. 16 Represent desired and actual path



(a)

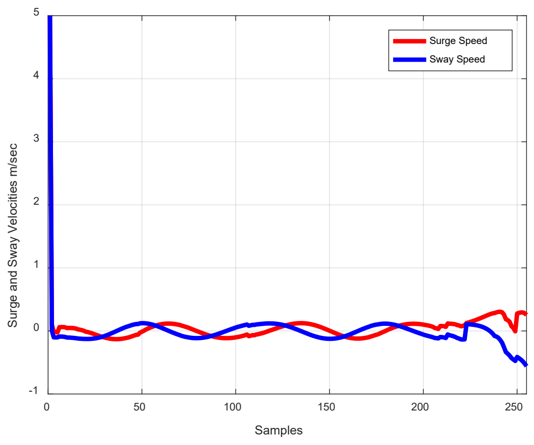


(b)

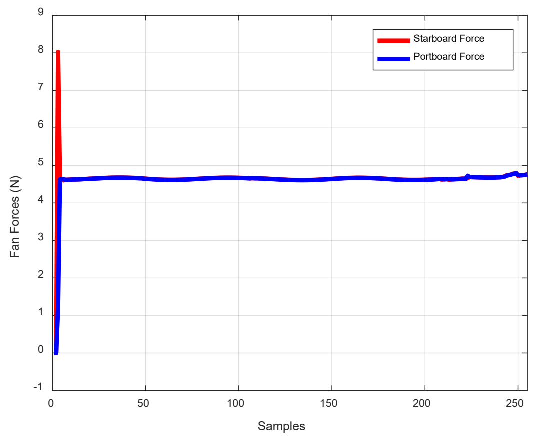
Figure. 17 Scenario 2 result: (a) the linear velocity of the hovercraft and (b) the hovercraft orientation

In order to demonstrate the effectiveness of the ABC-SPPSO hybrid algorithm in finding the shortest path, a comparative study was done with other researchers who utilized other algorithms in a static environment. Initially, the hybrid approach was evaluated against the hybrid firefly and the modified chaotic particle swarm optimization (FAMCPSO) algorithms in the identical congested environment described in [34], with a workspace measuring [600×800] cm. The simulation result, utilizing the hybrid ABCSP-PSO algorithm depicted in Fig. 19 (a) and (b), yields a path length of 649.58 cm at iteration 22.

Tables 9 and 10 show that the proposed hybrid method gives a path with a shorter distance and consumes less time in finding the best path compared to PSO, CPSO, MCPSO, FA, and HFAMCPSO in the literature [34].

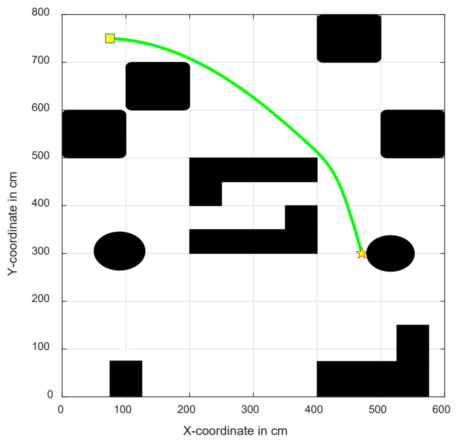


(a)

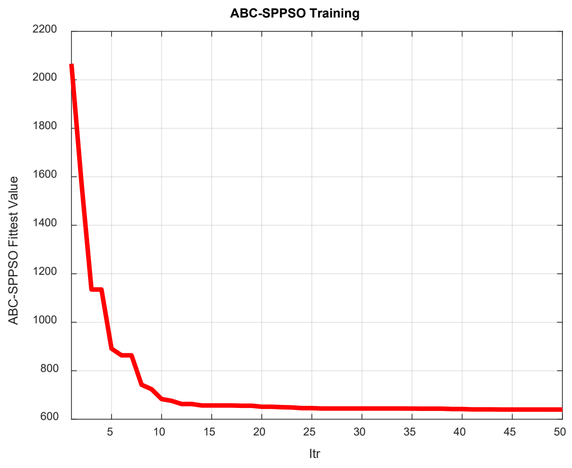


(b)

Figure. 18 Scenario 2 result: (a) surge and sway velocity of the hovercraft and (b) the starboard and the portboard fan forces.



(a)



(b)

Figure. 19 Comparison with [34]: (a) simulation result of the shortest path of the hybrid (ABC-SPPSO) algorithm and (b) the cost function

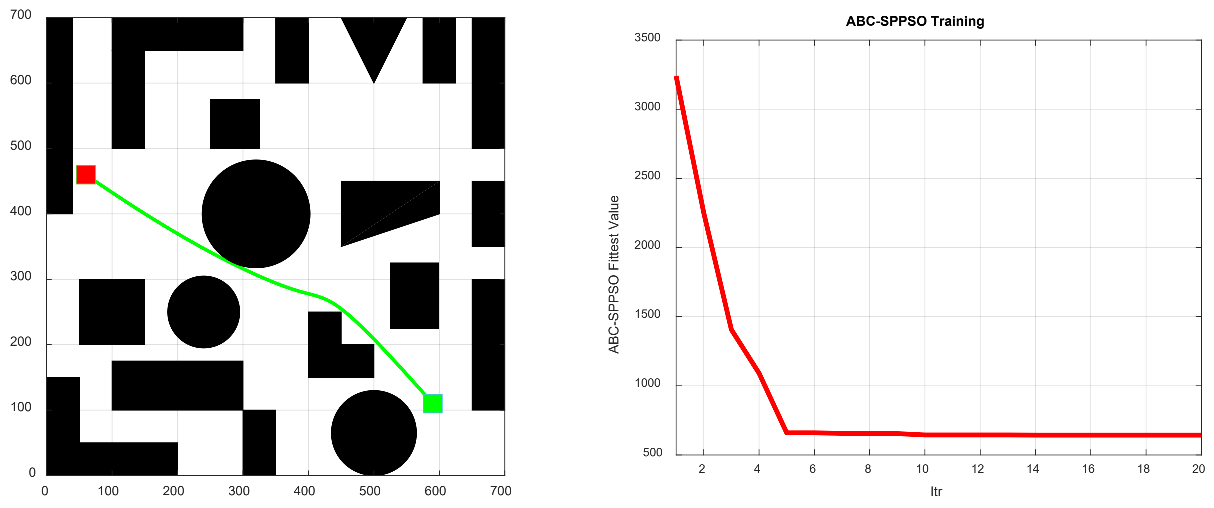
Tables 9 and 10 show that the proposed hybrid method gives a path with a shorter distance and consumes less time in finding the best path compared to PSO, CPSO, MCPSO, FA, and HFAMCPSO in the literature [34]. The issue with the method in [34] lies in its weakness in selecting an optimal number of the Fireflies direction. In particular, it only considers a 2D Fireflies direction. However, better results can be achieved by increasing the directions over 5D. This restriction limits the effectiveness of the path in terms of consuming more time, where the increase in the number of Fireflies directions leads to an increase in the number of iterations.

Table 9. Comparison of the hybrid algorithm with the literature [34]

Algorithms	Best path distance(cm)	Iterations
PSO [34]	750.834	73
CPSO [34]	739.168	60
MCPSO [34]	738.507	58
FA [34]	743.555	65
HFAMCPSO [34]	737.399	50
ABC-SPPSO	649.58	22

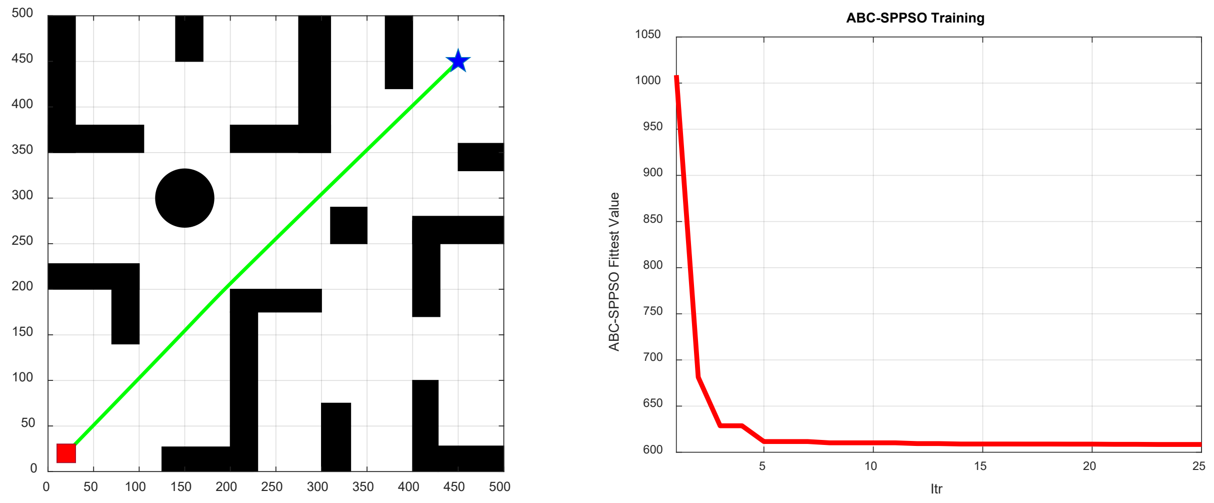
Table 10. Enhancement comparison of the hybrid algorithm with the literature [34]

Algorithms	Enhancement in distance	Enhancement in iteration
PSO	13.48%	69%
CPSO	12.12%	63.33%
MCPSO	12.04%	62.06%
FA	12.63%	66.15%
HFAMCPSO	11.90%	56%



(a) (b)

Figure. 20 A comparison with [35]: (a) simulation result of the shortest path of the hybrid (ABC-SPPSO) algorithm and (b) the cost function



(a) (b)

Figure. 21 A comparison with [11]: (a) simulation result of the shortest path of the hybrid (ABC-SPPSO) algorithm and (b) the cost function

Furthermore, the ABC-SPPSO algorithm was compared to the QOPSO algorithm proposed in [35]. The comparison was conducted in a static environment with a workspace measuring [700×700] cm. The starting point was located at coordinates (60, 460) cm and the target point was located at coordinates (590, 110) cm.

Figure 20 (a) and (b) displays the outcomes of the simulation conducted in the environment using the ABC-SPPSO algorithm. The algorithm generates a path with a length of 644.69 cm at the 10th iteration.

Furthermore, the ABC-SPPSO algorithm was compared to the QOPSO algorithm proposed in [35]. The comparison was conducted in a static environment with a workspace measuring [700×700] cm. The starting point was located at coordinates (60, 460) cm and the target point was located at coordinates (590, 110) cm.

Figure 20 (a) and (b) displays the outcomes of the simulation conducted in the environment using the ABC-SPPSO algorithm. The algorithm generates a path with a length of 644.69 cm at the 10th iteration. The comparative analysis and the enhancement percentages are presented in Table 11,

Table 11. A comparison of the hybrid algorithm with the literature [35]

Algorithms	The best path distance(cm)	Iterations
QO [35]	659.420	-
PSO [35]	757.1048	20
QOPSO [35]	657.1271	12
Proposed hybrid ABC-SPPSO	644.69	10

Table 12. Enhancement comparison of hybrid algorithm with literature [35]

Algorithms	Enhancement in distance	Enhancement in iteration
QO	2.23%	-
PSO	14.84%	50%
QOPSO	1.89%	16.66%

Table 13. Comparison of the hybrid algorithm with the literature [11]

Algorithms	Best path distance(cm)	Iterations
RRT [11]	665.4	-
RRT* [11]	649.6	-
PSO [11]	660	15
RRT*PSO [11]	615.3	12
ABC-SPPSO	610.21	8

Table 14. Enhancement comparison of the hybrid algorithm with the literature [11]

Algorithms	Enhancement in distance	Enhancement in iteration
RRT	8.29%	-
RRT*	6.06%	-
PSO	7.54%	46.66%
RRT*PSO	0.82%	33.33%

and Table 12 demonstrates the efficacy of our suggested hybrid algorithm in comparison to the Quarter Orbit and the PSO algorithms. In [35], the number of orbits is not ideal because as the number of orbits increases, the path becomes shorter and smoother. As a result, one of the limitations of this strategy is that more iterations are required when the number of orbits is not optimal.

Finally, the ABC-SPPSO algorithm was evaluated against RRT, RRT*, PSO, and the hybrid RRT*PSO algorithm proposed in [11]. The evaluation was conducted in a static environment with a workspace measuring [500×500] cm. The simulation results obtained using the ABC-SPPSO method are displayed in Fig. 21 (a) and (b). The algorithm produces a path with a length of 610.21 cm at the 8th iteration.

Tables 13 and 14 show the comparison results of our proposed ABC-SPPSO with RRT, RRT*, PSO, and RRT*PSO, confirming the effectiveness of the proposed algorithm in requiring less time to find the best path. In [11], due to generating a limited number of random points, there were no more than 14, where the quality of the path improves as the quantity of random points increases. However, this necessitates a substantial number of iterations.

6. Conclusion

This study presented a hybrid method designed to address three problems, namely (1) reaching the target without hitting any obstacles, (2) producing a path with the minimum distance to the goal, and (3) achieving the smoothest trajectory. A stochastic approach was suggested for resolving the pathfinding issue for the hovercraft. This technique was employed to identify the optimal path within an obstructed environment. In particular, the hybrid method combines the strengths of both the Artificial Bee Colony and the SP-PSO algorithms. Based on the MATLAB simulation results, it was evident that the proposed hybrid algorithm outperforms the original algorithms in finding the optimal path for the hovercraft in obstacle-filled environments in two scenarios (with different start and target points). In the first scenario, the hybrid algorithm enhanced the path length compared to ABC by (2.71%) in terms of distance and by (75.55%) in terms of iterations, and compared to SP-PSO, it enhanced the path by (24.17%) in terms of distance and by (60.71%) in terms of iterations. In the second scenario, the hybrid algorithm enhanced the path length compared to ABC by (2.01%) in terms of distance and by (78.26%) in terms of iterations. Furthermore, compared to SP-PSO, it enhances the path by (23.05%) in terms of distance and by (33.33%) in terms of iterations. Then, the hybrid algorithm was compared to the hybrid firefly algorithm modified with chaotic particle swarm optimization (HFAMCPSO), and it enhanced the path length by (11.90%) in terms of distance and (56%) in terms of iterations. Subsequently, the hybrid algorithm was further compared to the Quarter Orbit combined with particle swarm optimization technique, and the proposed method enhanced the distance to reach the goal by (1.89%) and by (16.66%) in terms of iterations. Finally, the hybrid algorithm was compared with the rapidly random tree star particle swarm optimization (RRT*PSO), achieving enhancement in distance and iterations by (0.82%)

and (33.33%), respectively. After considering the hovercraft's path length in comparison to other studies, we found that the proposed hybrid ABC-SPPSO algorithm produced the best combination of the shortest path and collision avoidance with fewer iterations to obtain the best path because the proposed algorithm has high convergence speed and it achieves a trade-off between exploration and exploitation.

As a future endeavour, we recommend adapting the proposed hybrid algorithm to function effectively in a dynamic environment.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Sura Muhi Husssein and Ahmed Sabah Al-Araji enhanced and developed an intelligent path-finding algorithm for the hovercraft model. Sura Muhi Husssein described the proposed algorithm for path finding. Ahmed Sabah Al-Araji explained the kinematic and dynamic hovercraft model. The two authors discussed the proposed simulation results of this work.

References

[1] H. Karami and R. Ghasemi, "Adaptive Neural Observer-Based Nonsingular Super-Twisting Terminal Sliding-Mode Controller Design for a Class of Hovercraft Nonlinear Systems", *Journal of Marine Science and Application*, Vol.20, No.2, pp.325-332, 2021.

[2] D. Lu, W. Xie, D. Cabecinhas, R. Cunha, and C. Silvestre, " Path Following Controller Design for an Underactuated Hovercraft with External Disturbances", In: *Proc. of International Conf. on Control*, Automation and Systems, Jeju, Korea, pp.76-81, 1996.

[3] G. E. M. Abro, Z. A. Ali, and B. Jabeen, "Prototyping Non-holonomic Hovercraft for Path Planning and Obstacle Avoidance", *Sir Syed University Research Journal of Engineering and Technology*, Vol.9, No.2, pp.1-6, 2019.

[4] Z. A. Ahmed, and S. M. Raafat, "Enhanced Global Path Planning Efficiency by Bidirectional A*, Gradient Descent, and Orientation Interpolation Algorithms", *International Journal of Intelligent Engineering and Systems*, Vol.16, No.5, pp.449-461, 2023, doi: 10.22266/ijies2023.1031.39.

[5] K. E. Dagher and M. N. Abdullah, "Airborne Computer System Based Collision-Free Flight Path Finding Strategy Design for Drone Model", *International Journal of Intelligent Engineering and Systems*, Vol.14, No.6, pp.234-248, 2021, doi: 10.22266/ijies2021.1231.22.

[6] D. Cabecinhas, P. Batista, P. Oliveira, and C. Silvestre "Hovercraft Control With Dynamic Parameters Identification", *IEEE Transactions on Control Systems Technology*, Vol.26, No.3, pp.785-796, 2017.

[7] A. P. Aguiar, L. Cremean, and J. P. Hespanha, "Position Tracking for a Nonlinear Under-actuated Hovercraft: Design and Experimental Results", In: *Proc. of 42nd IEEE Conference on Decision and Control*, pp.3858-3863, 2003.

[8] S. K. Debnath, R. Omar, S. Bagchi, S. E. N, M. H. A. S. Kandar, K. Foyso, and T. K. Chakraborty, "Different Cell Decomposition Path Planning Methods for Unmanned Air Vehicles - A Review", In: *Proc. of the 11th National Technical Seminar on Unmanned System Technology*, pp.99-111, 2021.

[9] H. Choset, S. Walker, K. E. Ard, and J. Burdick, "Sensor-Based Exploration: Incremental Construction of the Hierarchical Generalized Voronoi Graph", *International Journal of Robotics Research*, Vol.19, No.2, pp.96-125, 2000.

[10] P. Wang, S. Gao, L. Li, B. Sun, and S. Cheng, "Obstacle Avoidance Path Planning Design For Autonomous Driving Vehicles Based On An Improved Artificial Potential Field Algorithm", *Energies,* Vol.12, No.12, pp.1-14, 2019.

[11] A. A. A. Rasheed, A. S. Araji, and M. N. Abdullah, "Static and Dynamic Path Planning Algorithms Design for a Wheeled Mobile Robot Based on a Hybrid Technique", *International Journal of Intelligent Engineering and Systems*, Vol.15, No.4, pp.161-175, 2022, doi: 10.22266/ijies2022.0831.16.

[12] M. Badamashi, M. F. Emzir, S. El-ferik, and K. A. Sattar, "Autonomous Mobile Robot Path Planning Techniques, A Review (Part 1): Classical and Heuristic methods", *IEEE Access,* Vol.4, No.1, pp.1-15, 2023.

[13] E. W. Dijkstra, "A Note on Two Problems in Connexion with Graphs", *Numerische Mathematik*, Vol.1, No.1, pp.269-271, 1959.

[14] O. A. Wahhab and A. S. Araji, "Path Planning and Control Strategy Design for Mobile Robot Based on Hybrid Swarm Optimization Algorithm", *International Journal of Intelligent Engineering and Systems*, Vol.14, No.3, pp.565-579, 2021, doi: 10.22266/ijies2021.0630.48.

[15] A. Stentz, *Optimal and Efficient Path Planning for Unknown and Dynamic Environments*, Carnegie-Mellon University Pittsburgh Pa Robotics, 1993.

[16] I. Darwin, and S. Liawatimena, "Dynamic Map Pathfinding Using Hierarchical Pathfinding Theta-Star (HPT*) Algorithm", *International Journal of Engineering and Emerging Technology*, Vol.6, No.2, pp.80-85, 2021.

[17] F. Gul, W. Rahiman, S. S. N. Alhady, A. Ali, I. Mir, and A. Jalil ," Meta-heuristic approach for solving multi-objective path planning for autonomous guided robot using PSO–GWO optimization algorithm with evolutionary programming", *Journal of Ambient Intelligence and Humanized Computing*, Vol.12, No.1, pp.7873-7890., 2021.

[18] X. Yuan, X. Yuan, and X. Wang "Path Planning for Mobile Robot Based on Improved Bat Algorithm", *Sensors,* Vol.21, No.13, pp.1-14, 2021.

[19] E. K. Elsayed, A. H. Omar, and K. E. Elsayed, "Smart Solution for STSP Semantic Traveling Salesman Problem via Hybrid Ant Colony System with Genetic Algorithm", *International Journal of Intelligent Engineering and Systems*, Vol.13, No.5, pp.476-489, 2020, doi: 10.22266/ijies2020.1031.42.

[20] V. Jamshidi, V. Nekoukar, and M. H. Refan, "Real Time Uav Path Planning By Parallel Grey Wolf Optimization With Align Coefficient On Can Bus", *Cluster Computing*, Vol.24, No.3, pp.2495–2509, 2021.

[21] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A Comprehensive Survey: Artificial Bee Colony (ABC) Algorithm and Applications", *Artificial Intelligence Review*, Vol.42, No.1, pp.21–57, 2012.

[22] A. A. Kareem, B. K. Oleiwi, and M. J. Mohamed, "Planning the Optimal 3D Quadcopter Trajectory Using a Delivery System-Based Hybrid Algorithm", *International Journal of Intelligent Engineering and Systems*, Vol.16, No.2, pp.427-439, 2023, doi: 10.22266/ijies2023.0430.34.

[23] E. I. Abba, S. D. Hasan, and R. Jawad, "Optimum Path Finding From Multi-Paths Based On Fuzzy Logic System", *International Journal of Engineering Applied Sciences and Technology***,** Vol.5, No.4, pp.39-46, 2020**.**

[24] Y. Zhu, "Using neural networks to explore path planning algorithms for robots", In: *Proc of the 3rd International Conference on Signal Processing and Machine Learning*, pp.566-572, 2023.

[25] H. Sun, W. Zhang, R. Yu, and Y. Zhang, "Motion Planning for Mobile Robots_Focusing on Deep Reinforcement Learning: A Systematic Review", *IEEE Access*, Vol.9, No.1, pp.69061–69081, 2021.

[26] D. Karaboga, "An Idea Based on Honey Bee Swarm for Numerical Optimization", *Technical Report-TR06*, Vol.200, No.1, pp.1-10, 2005.

[27] F. Harfouchi, H. Habbi, C. Ozturk, and D. Karaboga, "Modified Multiple Search Cooperative Foraging Strategy for Improved Artificial Bee Colony Optimization with Robustness Analysis", *Soft Computing*, Vol.22, No.1, pp.6371-6394, 2017.

[28] S. F. Raheem, and M. Alabbas, "Dynamic Artificial Bee Colony Algorithm with Hybrid Initialization Method", *Informatica*, Vol.45, No.6, pp.103-114, 2021.

[29] C. Wang, P. Shang, and P. Shen, "An Improved Artificial Bee Colony Algorithm based on Bayesian Estimation", *Complex & Intelligent Systems,* Vol.8, No.6, pp.4971-4991, 2022.

[30] J. Kennedy and R. Eberhart, "Particle Swarm Optimization", In: *Proc. of International Conference on Neural Networks*, pp. 1942-1948, 1995.

[31] E. Krell, A. Sheta, A. P. R. Balasubramanian, and S. A. King, "Collision-Free Autonomous Robot Navigation In Unknown Environments Utilizing Pso For Path Planning", *Journal of Artificial Intelligence and Soft Computing Research*, Vol.9, No.4, 2019.

[32] S. I. A. Meerza, M. Islam, and M. M. Uzzal, "Optimal Path Planning Algorithm for Swarm of Robots using Particle Swarm Optimization Technique", In: *Proc of 3rd International Conference on Information Technology*, Information System and Electrical Engineering (ICITISEE). IEEE, pp.330–334, 2018.

[33] P. P. Dash, and D. Patra, "Mutation Based Self Regulating and Self Perception Particle Swarm Optimization for Efficient Object Tracking In a Video", *Measurement,* Vol.144, No.1, pp.311-327, 2019.

[34] N. A. K. Zghair, and A. S. Araji, "Intelligent Hybrid Path Planning Algorithms for Autonomous Mobile Robots", *International Journal of Intelligent Engineering and Systems*, Vol.15, No.5, pp.309-325, 2022, doi: 10.22266/ijies2022.1031.28.

[35] Z. E. Kanoon, A. S. Araji, and M. N. Abdullah, "Enhancement of Cell Decomposition Path-Planning Algorithm for Autonomous Mobile Robot Based on an Intelligent Hybrid

Received: December 29, 2023. Revised: January 19, 2024.

Optimization Method", *International Journal of Intelligent Engineering and Systems*, Vol.15, No.3, pp.161-175, 2022, doi: 10.22266/ijies2022.0630.14.